

מספר ת"ז
מספר מחברת

מועד הבחינה :
קיץ תשפ"ה - 2025 – מועד א'
מספר השאלון: 97103
מילון עזר

אלגוריתמיקה ותכנות

הנדסאים וטכנאים – הנדסת תוכנה

הנחיות לבחינה

א. משך הבחינה: ארבע שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה:

חלק א' – 36 נקודות

שאלות 1–4: יש לענות על שלוש שאלות בלבד. ערך כל שאלה – 12 נקודות.

חלק ב' – 30 נקודות

שאלות 5–7: יש לענות על שתי שאלות בלבד. ערך כל שאלה – 15 נקודות.

חלק ג' – 34 נקודות

שאלות 8–10: יש לענות על שתי שאלות בלבד. ערך כל שאלה – 17 נקודות.

בסך הכול: 100 נקודות.

ג. חומר עזר 1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית).

2. מותר לשימוש: קלסר אחד בלבד עם חומר ההרצאות. אין להוציא דפים מהקלסר.

אין לצרף ספרים או חוברות עם פתרונות.

ד. הוראות כלליות: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות המבחן, ולוודא שהן מובנות.

2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת המבחן).

3. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי סדר כתיבתן בעמוד זה. לא ייבדקו תשובות עודפות.

4. יש לכתוב את התשובות במחברת הבחינה בעט בלבד, בכתב יד ברור.

5. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה.

6. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק.

7. יש להציג פתרון מלא ומנומק, כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד.

8. יש להסביר בפירוט כל תוכנית שנכתבה, תוכנית ללא הסבר מפורט לא תזכה בניקוד.

9. אם לדעתכם חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר להמשיך בפתרון השאלה. נמקו את בחירתכם.

חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!

ההנחיות בשאלון זה מנוסחות בלשון זכר, אך מכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

הנחיות כלליות לנבחנים ב-Python

1. דוגמה להוראה לקליטת מספר שלם :

```
x = int(input());
```

2. דוגמה להוראה לקליטת מספר עשרוני :

```
y = double(input());
```

3. דוגמה להוראה לקליטת מחרוזת :

```
str = input();
```

4. יצירת מספר שלם אקראי num בין X ל-Y כולל ($X \leq \text{num} \leq Y$)

```
import random
```

```
...
```

```
num = random.randint(x, y)
```

חלק א'

ענו על שלוש מבין השאלות 1–4 (ערך כל שאלה – 12 נקודות).

שאלה 1

בשיטת ההצפנה "סטופיד" משתמשים כדי להצפין מסר סודי ללא שימוש בכלים ואלגוריתמים מתקדמים. לשיטה שלושה שלבים:

- הסרת רווחים מהמסר המקורי.
- החלפת סדר התווים בכל זוג תווים צמודים (אם מספר התווים אי-זוגי, התו האחרון נשאר ללא שינוי).
- כתיבת התווים מהסוף להתחלה (הפיכת סדר התווים).

לדוגמה:

אם הטקסט המקורי היה "Hello World!" אז הטקסט המוצפן יהיה "!ldoroWllHe"
Hello World! → HelloWorld! → eHllWorodl! → !ldoroWllHe

כתבו פונקציה אשר מקבלת מחרוזת ומחזירה מחרוזת מוצפנת.

- Split לפי רווח
- בדיקה החלפת מקום בלולאת FOR
- כתיבה מה סוף להתחלה

```
- def stupid_encrypt(msg):  
    msg = msg.replace(" ", "")  
  
    chars = list(msg)  
    for i in range(0, len(chars) - 1, 2):  
        chars[i], chars[i + 1] = chars[i + 1], chars[i]  
  
    return ''.join(chars[::-1])  
  
def stupid(msg):  
    new_msg=""  
    msg=msg.replace(" ", "")  
    for i in range (0, len(msg)-1, 2):  
        new_msg+=msg[i+1]+msg[i]  
    if len(msg)%2==1:  
        new_msg+=msg[-1]  
    return new msg[::-1]
```

שאלה 2

נתונה רשימה arr בגודל n המכילה מספרים שלמים שונים זה מזה.

נגדיר "נקודת איזון" כאינדקס k ברשימה, כך שמכפלה כל האיברים משמאל לאינדקס k שווה למכפלה כל האיברים מימין לאינדקס k (לא כולל את האיבר באינדקס k).

כתבו פונקציה שמקבלת רשימה של מספרים שלמים ומחזירה את האינדקס של נקודת האיזון הראשונה מצד שמאל. אם לא קיימת נקודת איזון, הפונקציה תחזיר -1.

```
def find_balnce(arr,k):
    left=1
    right=1
    for i in range(0,k):
        left*=arr[i]
    for i in range(k+1,len(arr)):
        right*=arr[i]
    return left==right

def balance(arr):
    for i in range(1,len(arr)-1):
        if find_balnce(arr,i):
            return i
    return -1
```

שאלה 3

נתונה הפונקציה what הבאה:

```
def what(arr):
    c = 0
    m = 0
    for i in range(len(arr)):
        if arr[i] % 2 == 0:
            c += 1
        else:
            if c > m:
                m = c
            c = 0
    if c > m:
        return c
    return m
```

(4 נק') א. נתונה רשימה [12, 6, 3, 17, 4, 5, 2, 8, 10, 13].arr =

עקבו אחרי זימון what(arr) ורשמו מה תהיה תוצאת הזימון. יש להראות מעקב אחרי ביצוע!

תוצאת הזימון היא 3

c	m	Arr[i]	arr[i] % 2==0	c>m
0	0			
1	0	12	True	
2	0	6	true	
0	2	3	false	true
0	2	17	false	false
1	2	4	true	
0	2	5	false	False
1	2	2	true	
2	2	8	true	
3	2	10	true	
0	3	13	false	false
				false

(3 נק') ב. תנו דוגמה לרשימה `arr` בגודל **שישה** איברים הכוללת מספרים שלמים שונים זה מזה, שעבורו הפונקציה `what` תחזיר ערך **ארבע**.

`Arr=[1,2,4,6,8,3]`

(3 נק') ג. מהו הערך **הקטן ביותר** שהפונקציה `what` יכולה להחזיר? תנו דוגמה לרשימה `arr` בגודל **שישה** איברים הכוללת מספרים שלמים שונים זה מזה, שעבורו הפונקציה `what` תחזיר ערך זה.

הערך הקטן ביותר הוא 0,

`arr=[1,3,5,7,9,11,13]` לדוגמא

(2 נק') ד. מה מבצעת שהפונקציה `what` באופן כללי? מחזירה את מספר(כמות) האיברים הזוגיים הרצופים הגדול במערך.

שאלה 4

במפעל יש שני סוגים של עובדים: פועלים, ומהנדסים. שכרו של כל עובד מחושב כך:

- עבור כל אחת מ-160 שעות הראשונות פועל מקבל 80 ₪ לשעת העבודה ומהנדס – 150 ₪.
- עבור כל שעה נוספת מקבל העובד תוספת של 70 ₪.

נתונה מחלקה בשם Employee עם התכונות הבאות:

- id - מחרוזת המייצגת את תעודת הזהות של העובד.
- status - מספר שלם המייצג את סוג העובד (1 עבור מהנדס, 2 עבור פועל).
- hours - מספר שלם המייצג את סך שעות העבודה של העובד בחודש.

```
class Employee:
```

```
    def __init__(self, id, status):
        self.id = id
        self.status = status
        self.hours = 0
```

במחלקה הוגדרו כל הפעולות set ו-get וגם הפונקציה `__str__`.

(3 נק') א. כתבו במחלקה Employee פונקציה `get_salary()` לחישוב משכורת חודשית של עובד. כותרת הפונקציה:

```
def get_salary (self)
```

```
def get_salary(self):
    base_hours = min(self.hours, 160)
    extra_hours = max(0, self.hours - 160)
    rate = 150 if self.status == 1 else 80
    return base_hours * rate + extra_hours * 70
```

```
def get_salary(self):
    base_hours = 160 if self.hours < 160 else self.hours
    extra_hours = 0 if self.hours - 160 < 0 else self.hours - 160
    rate = 150 if self.status == 1 else 80
    return base_hours * rate + extra_hours * 70
```

(3 נק') ב. לפניכם קטע תכנית לקליטת נתוני שכר של עובד מסוים. עבור כל אחד מ-25 ימי העבודה בחודש ייקלטו שעת תחילת העבודה ושעת סיום העבודה. אחרי קליטת נתונים קטע מדפיס את שכרו של העובד.

השלימו את החסר

```
n = input("תזוהות תעודת:")

t = int(input("העובד סוג (1=מהנדס, 2=פועל):"))

emp = emp = Employee(n, t)

s = 0

for i in range(25):
```

שאלון 97103, קיץ תשפ"ה – 2025 – מועד א'

```
a = int(input("שעת התחלה"))
b = int(input("שעת סיום"))
s = b - a

emp.set_hours(s)
print(emp)
```

(6 נק') ג. כתבו פונקציה אשר מקבלת רשימת של אובייקטים המכילים את נתוני השכר של כל עובדי המפעל. על הפונקציה להדפיס מספרי תעודות הזהות של הפועלים אשר משכורתם גבוהה מהממוצע משכורת של מהנדסים.

```
def above_avg_workers(workers):
    engineers = [e.get_salary() for e in workers if e.status == 1]
    if not engineers:
        return
    avg = sum(engineers) / len(engineers)
    for e in workers:
        if e.status == 2 and e.get_salary() > avg:
            print(e.id)
```

חלק ב'

ענו על שתיים מבין השאלות 5–7 (ערך כל שאלה – 15 נקודות).

שאלה 5

הגדרה:

מספר ראשוני הוא מספר שלם חיובי שמתחלק בלי שארית רק ב-1 ובעצמו.

4 נקי) א. כתבו פונקציה המקבלת מספר שלם וחיובי ובודקת אם הוא ראשוני. אם כן – הפונקציה תחזיר ערך

True, ולא – הפונקציה תחזיר ערך False.

כותרת הפונקציה:

```
def is_primary(num)
```

השערת גולדבך (Goldbach) היא השערה בתורת המספרים, שלפיה כל מספר זוגי ניתן להציג כסכום של שני

מספרים ראשוניים.

לדוגמא:

$$3+5 = 8 \text{ (או } 1+7)$$

$$5+11 = 16 \text{ (או } 3+13)$$

$$5+19 = 24 \text{ (או } 7+17 \text{ או } 1+23 \text{ או } 11+13)$$

$$3+43 = 46 \text{ (או } 5+41 \text{ או } 23+23)$$

4 נקי) ב. כתבו פונקציה המקבלת מספר שלם זוגי num. הפונקציה תחזיר את מספר הזוגות של מספרים

ראשוניים אשר סכומם שווה למספר num.

כותרת הפונקציה:

```
def count_primary_pairs(num)
```

4 נקי) ג. כתבו פונקציה המקבלת מספר שלם זוגי num. הפונקציה תחזיר רשימה (list) של כל זוגות

המספרים הראשוניים שסכומם שווה לו. כל זוג ייוצג כטאפל (tuple) או כרשימה קטנה יותר בתוך

הרשימה הגדולה.

לדוגמא:

עבור המספר 42 הפונקציה תחזיר את הרשימה הבאה:

```
[(1, 41), (5, 37), (11, 31), (13, 29), (19, 23)]
```

כי:

$$42 = 41 + 1$$

$$42 = 37 + 5$$

$$42 = 31 + 11$$

$$42 = 29 + 13$$

$$42 = 23 + 19$$

כותרת הפונקציה:

```
def all_primary_pairs(num)
```

```
def is_primary(num):  
    for i in range(2,num//2):
```

```

        if num%i==0:
            return False
        return True

def count_primary_pairs(num):
    count=0
    for i in range(num//2):
        if is_primary(i) and is_primary(num - i):
            count += 1
    return count

def all_primary_pairs(num):
    pairs = []
    for i in range(1, num // 2):
        j = num - i
        if is_primary(i) and is_primary(j):
            pairs.append((i, j))
    return pairs

```

3 נק' ד. מהי סיבוכיות זמן הריצה של כל הפעולות שכתבתם? הסבירו את תשובתכם.

$O(\log n)$

$O(n \log n)$

$O(n \log n)$

שאלה 6

מפעל לייצור דגלים של מדינות שונות מעוניין במערכת ניהול מלאי ממוחשבת.

לשם כך הוגדרה המחלקה "דגל" (Flag) ולה תכונות הבאות:

- שם מדינה (country) - מחרוזת (str)
- אורך (length) - מספר שלם (int)
- רוחב (width) - מספר שלם (int)

במחלקה הוגדרה פונקציה בונה (`__init__`) המקבלת ערכים לכל התכונות ופעולות `get/set` (מאפיינים) והפעולה `__str__`

3 נק' א. הגדירו מחלקה בשם "מחסן דגלים" (Warehouse) הכוללת את התכונות הבאות:

- רשימה של דגלים (flags) - רשימה של עצמים מסוג "דגל" (Flag) בגודל מקסימלי של 100.
- רשימה של כמויות (quantities) - רשימה של מספרים שלמים (int) באותו הגודל כמו רשימת הדגלים, כאשר כל תא מציין את הכמות של הדגל המתאים ברשימת הדגלים (הדגל במקום ה-i ברשימת הדגלים מתאים לכמות במקום ה-i ברשימת הכמויות).
- מספר הדגלים הנוכחי במחסן (current_flags) - מספר שלם (int).

כתבו פונקציה בונה (`__init__`) למחלקה המאתחלת את רשימת הדגלים ואת רשימת הכמויות בגודל 100 ואת מספר הדגלים הנוכחי ל-0.

```

def __init__(self):
    self.flags = [None] * 100
    self.quantities = [0] * 100
    self.current_flags = 0

```

4 נק') ב. כתבו במחלקה "מחסן דגלים" פונקציה המקבלת שם מדינה, אורך ורוחב של דגל ומספר דגלים ומוסיפה את הדגל למחסן. אם כבר קיים דגל זהה במחסן, הפונקציה תעדכן את הכמות שלו. אם המחסן מלא, הפונקציה תדפיס הודעה מתאימה.
כותרת הפונקציה:

```
def add(self, country, length, width, quantity)
```

```
def add(self, country, length, width, quantity):
    if self.current_flags >= 100:
        print("מלא המחסן")
        return
    for i in range(self.current_flags):
        f = self.flags[i]
        if f.country == country and f.length == length and f.width == width:
            self.quantities[i] += quantity
        return

    self.flags[self.current_flags] = Flag(country, length, width)
    self.quantities[self.current_flags] = quantity
    self.current_flags += 1
```

4 נק') ג. כתבו במחלקה "מחסן דגלים" פונקציה מקבלת מספר שלם המייצג כמות מינימלית ומדפיסה את שמות פרטים של כל הדגלים במחסן שנמצאים בכמות הקטנה מהכמות המינימלית.
כותרת הפונקציה:

```
def flags_with_min_quantity(self, min_quantity)
```

```
def flags_with_min_quantity(self, min_quantity):
    for i in range(self.current_flags):
        if self.quantities[i] < min_quantity:
            print(self.flags[i])
```

4 נק') ד. לקראת ביקור ראשי מדינות שונות יש להכין דגלים לקישוט העיר. כתבו במחלקה "מחסן דגלים" מקבלת רשימה של שמות מדינות (רשימה של מחרוזות) ומדפיסה עבור כל מדינה ברשימה את כמות של כל הדגלים שלה הקיימים במחסן. שימו לב שבמחסן יכולים להיות דגלים שונים של אותה מדינה!
כותרת הפונקציה:

```
def print_flags(self, countries)
```

```
def print_flags(self, countries):
    for i in range(self.current_flags):
        if self.flags[i].country in countries:
            print(f"{self.flags[i]} - {self.quantities[i]} יחידות")
```

שאלה 7

נתונות ארבע הפעולות הבאות:

```
def one(num):
    return two(num, 0, 0)

def two(num, s, c):

    if num == 0:
        return s / c if c != 0 else 0.0
    return two(num // 10, s + (num % 10), c + 1)

def three(arr, p):
    n = len(arr)
    temp = [0] * n
    four(arr, temp, 0, 0, n - 1, p)
    return temp

def four(arr, temp, k, left, right, p):
    if k < len(arr):
        if one(arr[k]) > one(arr[p]):
            temp[right] = arr[k]
            four(arr, temp, k + 1, left, right - 1, p)
        else:
            temp[left] = arr[k]
            four(arr, temp, k + 1, left + 1, right, p)
```

5 נק' א.

1. עקבו אחרי זימון הפונקציה `one(6123)` ורשמו את תוצאת הזימון. **יש להראות מעקב!**

תחזיר 3.0.

2. תנו דוגמה של מספר שלם וחיובי גדול מ-100 שעבורו הפונקציה תחזיר ערך **חמש**.

2684

3. מה מבצעת הפונקציה `one(num)` עבור מספר שלם `num`?

מחזירה את ממוצע סכום הספרות

10 נק' ב.

נתונה הרשימה `a = [126, 9, 35, 2684, 8941, 24]`,

1. עקבו אחרי זימון הפונקציה `three(a, 3)` ושרמו מה תהיה תוצאת הזימון.

תשובה: `[9, 2684, 8941, 35, 126, 24]`

יש להראות מעקב אחרי ביצוע הפונקציה `three`, אין צורך להראות מעקב אחרי הפעולות `one` ו-`two`.

2. האם קיים ערך p שעבורו זימון הפונקציה $three(a, p)$ יגרום לשגיאה? הסבירו את תשובתכם.
אם P גדול מאורך הרשימה או P קטן מ-0
3. תנו דוגמה של רשימה arr בגודל 6 איברים כך שזימון הפונקציה $three(arr, p)$ תחזיר רשימה זהה ל- arr עבור כל ערך p תקין. אם הדבר אינו אפשרי, תסבירו למה.
אפשרי רק אם כל המספרים זהים ומחזירים את אותו ממוצע.
לדוגמא [11,11,11,11,11,11]
4. מה מבצעת הפונקציה $three(a, p)$ עבור רשימה של מספרים שלמים arr וערך שלם p .
יוצר רשימה חדשה שבה כל האיברים בתחילת הרשימה חישוב הממוצע ספרות שלהם קטן או שווה לאיבר שמופיע במקום P , שאר המספרים יופיעו בסוף הרשימה.

חלק ג'

ענו על שתיים מבין השאלות 8–10 (ערך כל שאלה – 17 נקודות).

שאלה 8

במשחק תפזורת (Word Search) הפותר צריך לחפש בתוך מטריצה (רשימה של רשימות) של אותיות, רצפים של אותיות המרכיבים את המילה הנתונה. כיוון קריאת המילים יכול להיות מימין לשמאל, משמאל לימין, מלמעלה למטה ומלמטה למעלה.
לדוגמא, המילה "HELP" מופיעה במטריצה הבאה שלוש פעמים:

```
[['A', 'H', 'E', 'L', 'P', 'K'],
 ['B', 'T', 'F', 'E', 'H', 'H'],
 ['R', 'A', 'D', 'S', 'E', 'E'],
 ['P', 'L', 'E', 'H', 'L', 'L'],
 ['K', 'Z', 'H', 'R', 'T', 'P']]
```

A	H	E	L	P	K
B	T	F	E	H	H
R	A	D	S	E	E
P	L	E	H	L	L
K	Z	H	R	T	P

כתבו פונקציה המקבלת מטריצה דו-ממדית של אותיות arr ומילה word. על הפונקציה לבדוק האם המילה מופיעה במערך (לפי כללים של המשחק תפזורת). אם כן – הפונקציה תחזיר ערך True, ולא – הפונקציה תחזיר ערך False.
כותרת הפונקציה:

```
def exist(arr, word)
```

```
def exist(arr, word):
    for row in arr:
        line = ''.join(row)
        if word in line or word[::-1] in line:
            return True
    for col in range(len(arr[0])):
        column=""
        for row in range(len(arr)):
            column += (arr[row][col])
        print(column)
        if word in column or word[::-1] in column:
            return True
    return False
```

שאלה 9

המערכת ארגונית לניהול חשבונות משתמשים כוללת שתי המחלקות : Account ו-Users .

למחלקה Account תכונות הבאות :

- שם משתמש - username, מסוג מחרוזת, str
 - סיסמה הנוכחית - current_pass, מסוג מחרוזת, str
 - סיסמאות קודמות - pass_history, מסוג רשימה של 10 מחרוזות, list[str]. המערך ישמור עד 10 סיסמאות הקודמות של החשבון.
- (2 נק') א. כתבו בנאי (__init__) למחלקה Account שמקבל שם משתמש וסיסמה ראשונית כפרמטרים. הבנאי יאתחל את username, יאחסן את הסיסמה הראשונית ב-current_pass ובמקום הראשון ברשימה pass_history.

```
def __init__(self, username, password):
    self.username = username
    self.current_pass = password
    self.pass_history = [password]
```

(4 נק') ב. במערכת קיימת דרישת הבטחה לסיסמאות: הסיסמה תקינה אם היא עונה לפחות שלושה מתוך חמישה התנאים :

- הסיסמה חייבת להיות באורך של לפחות 8 תווים.
- הסיסמה חייבת להכיל לפחות שתי אותיות גדולות ('A'-'Z')
- הסיסמה חייבת להכיל לפחות ספרה אחת ('0'-'9')
- הסיסמה חייבת להכיל לפחות תו מיוחד אחד (כל תו שאינו אות או ספרה).
- אין תווים זהים צמודים

כתבו במחלקה Account את הפונקציה is_password_valid המקבלת מחרוזת ובודקת אם היא יכולה להיות סיסמה.

```
def is_password_valid(self, password):
    conditions = 0
    if len(password) >= 8:
        conditions += 1
    sum=0
    for ch in (password):
        if ch.isupper():
            sum+=1
    if sum>=2:
        conditions += 1
    for ch in (password):
        if ch.isdigit():
            conditions += 1
            break
    for ch in (password):
        if not ch.isalnum():
            conditions += 1
            break
    found=False
    for i in range(len(password)-1):
        if password[i] == password[i + 1]:
            found=True
    if not found:
        conditions += 1
    return conditions>=3
```

4 נק') ג. כתבו במחלקה Account את הפונקציה update_password המקבלת כפרמטר סיסמה חדשה. הסיסמה החדשה צריכה להיות תקינה. אם היא לא תקינה, הפונקציה לא מבצעת עדכון ומדפיסה הודעה מתאימה. הסיסמה החדשה צריכה להיות גם שונה מסיסמה הנוכחית ומסיסמאות הקודמות. אם היא זהה, הפונקציה לא מבצעת עדכון ומדפיסה הודעה מתאימה. אם הסיסמה החדשה תקינה ואינה זהה לאף אחת מהסיסמאות הקודמות, הפונקציה תעדכן את current_pass לסיסמה החדשה, ותוסיף את הסיסמה החדשה לתחילת הרשימה pass_history, תוך כדי הזזת הסיסמאות הקיימות (אם הרשימה מלאה, הסיסמה הישנה ביות תיזרס.

```
def update_password(self, new_pass):
    if not self.is_password_valid(new_pass):
        print("תקינה לא סיסמה")
        return
    if new_pass in self.pass_history:
        print("קודם בשימוש כבר סיסמה")
        return
    self.current_pass = new_pass
    self.pass_history.insert(0, new_pass)
    if len(self.pass_history) > 10:
        self.pass_history.pop()
```

למחלקה Users שתי תכונות:

- חשבונות משתמשים – accounts, מסוג רשימה עצמים מסוג Account, בגודל מקסימאלי 100.
 - מספר משתמשים פעילים - num_of_accounts, מסוג מספר שלם, int
- 3 נק') ד. כתבו במחלקה Users את הפונקציה add_user המקבלת שם משתמש וסיסמה ראשונית כפרמטרים.

הפונקציה תבדוק תחילה האם קיים כבר חשבון עם שם המשתמש הזה. אם קיים או אין יותר מקום, הפונקציה לא תוסיף משתמש חדש ותדפיס הודעה מתאימה. אם שם המשתמש לא קיים והרשימה accounts אינה מלאה, הפונקציה תיצור אובייקט חדש מסוג Account, תוסיף אותו לרשימה accounts ותעדכן את num_of_accounts.

```
def add_user(self, username, password):
    if self.num_of_accounts >= 100:
        print("המשתמשים למכסת הגעת")
        return
    if any(acc.username == username for acc in self.accounts):
        print("קיים כבר משתמש שם")
        return
    new_acc = Account(username, password)
    self.accounts.append(new_acc)
    self.num_of_accounts += 1
```

7 נק') ה. עקב ריבוי ניסיונות פריצה מצד גורמים זרים, הוחלט להקשיח את דרישות האבטחה לסיסמאות של המשתמשים. בהתאם לכך:

- הציעו לפחות שני שיפורים נוספים לחיזוק הסיסמאות, אשר יתווספו לתנאים לסיסמה תקינה.
- כתבו פונקציה המאפשרת לקבל את רשימת המשתמשים הנדרשים לחיזוק הסיסמה ולצד כל משתמש את רשימת התנאים בהם סיסמתו הישנה אינה עומדת. **נא הדפיסו הודעה מתאימה!**

ממשו את השינויים הנדרשים בהתאם לשיפורים כפי שהצעתם במחלקה אחת או יותר. יש לתעד היטב את הצעתכם.

```
def is_password_valid(self, password: str):
    issues = []

    if len(password) < 8:
        issues.append("length")
    if sum(1 for c in password if c.isupper()) < 2:
        issues.append("uppercase")
    if not any(c.isdigit() for c in password):
        issues.append("digit")
    if not any(not c.isalnum() for c in password):
        issues.append("special")
    if any(password[i] == password[i + 1] for i in range(len(password) - 1)):
        issues.append("repeated")
    if self.username.lower() in password.lower():
        issues.append("username_in_password")
    if not any(c.islower() for c in password):
        issues.append("lowercase")

    return (len(issues) <= 2), issues
```

```
def get_users_with_weak_passwords(self):
    print("שיפור הדורשות סיסמאות עם משתמשים:")
    any_weak = False
    for acc in self.accounts:
        valid, issues = acc.is_password_valid(acc.current_pass)
        if not valid:
            any_weak = True
            print(f"- {acc.username}: חלשה סיסמה ({', '.join(issues)})")
    if not any_weak:
        print("חלשות סיסמאות עם משתמשים אין.")
```

שאלה 10

נתונות ארבע הפעולות what ו- secret,secret ו- why ו- mystery.

```
def secret(a, s):
    s %= len(a)
    mystery(a, 0, len(a) - 1)
    mystery(a, 0, s - 1)
    mystery(a, s, len(a) - 1)
```

```
def mystery(a, start, end):
    while start < end:
        temp = a[start]
        a[start] = a[end]
        a[end] = temp
        start += 1
        end -= 1
```

```
def what(a, d):
    why(a, d, 0)
```

```
def why(a, d, i):
    if i < len(a):
        j = (i + d) % len(a)
        rest = a[j]
        why(a, d, i + 1)
        a[i] = rest
```

נתונה הרשימה arr מספרים שלמים הבאה:

[10, 3, 6, 8, 2, 11, 5]

0	1	2	3	4	5	6
10	3	6	8	2	11	5

3 נק' א. עקבו אחרי זימון הפונקציה `mystery(arr, 1, 4)` ורשמו מה יהיה תוכן הרשימה אחרי הזימון.

[10, 2, 8, 6, 3, 11, 5]

3 נק' ב. עקבו אחרי זימון הפונקציה `secret(arr, 3)` ורשמו מה יהיה תוכן הרשימה אחרי הזימון.

[2, 11, 5, 10, 3, 6, 8]

2 נק' ג. מה מבצעת הפונקציה `secret(a, d)` עבור רשימה של מספרים שלמים a ומספר שלם וחיובי d?

רוטציה ימינה של הרשימה ב d איברים

3 נק' ד. עקבו אחרי זימון הפונקציה `what(arr, 3)` ורשמו מה יהיה תוכן הרשימה אחרי הזימון.

יש להראות ערכי המשתנים בכל זימון הרקורסיבי

[8, 2, 11, 5, 10, 3, 6]

3 נק' ה. אחר זימון הפונקציה `what(brr, 2)` התקבלה הרשימה `brr` הבאה:

[10, 20, 30, 40, 50, 60]

0	1	2	3	4	5
10	20	30	40	50	60

מה היה תוכן של הרשימה `brr` לפני זימון הפונקציה?

`brr=[50,60,10,20,30,40]`

3 נק' ו. מה מבצעת הפונקציה `what(a, d)` עבור רשימה של מספרים שלמים a ומספר שלם וחיובי d?

רוטציה שמאלה של D מספרים.